

1. (ORIGINAL) A method of loading data into a data store connected to a computer, the method comprising the steps of  
identifying memory constraints;  
identifying processing capabilities; and  
determining a number of load and sort processes to be started in parallel based on the identified memory constraints and processing capabilities.

Sub  
B1 } 2. (ORIGINAL) The method of claim 1, further comprising determining a number of build processes based on the number of sort processes.

3. (ORIGINAL) The method of claim 1, wherein the number of sort processes does not exceed a number of indexes to be built.

4. (ORIGINAL) The method of claim 1, wherein the number of load processes does not exceed a number of partitions to be loaded.

5. (ORIGINAL) The method of claim 1, wherein the total number of load and sort processes does not exceed processing capabilities.

6. (ORIGINAL) The method of claim 1, wherein the memory utilized by the load and sort processes does not exceed memory constraints.

A1 7. (CURRENTLY AMENDED) The method of claim 1, wherein the number of load processes and the number of sort processes each require different processing power.

8. (CURRENTLY AMENDED) The method of claim 1, wherein the number of load processes and the number of sort processes each require similar processing power.

9. (CURRENTLY AMENDED) The method of claim 1, wherein the number of load processes is not equal to the number of sort processes.

Cost  
A1

10. (CURRENTLY AMENDED) The method of claim 1, wherein the number of load processes is equal to the number of sort processes.

11. (ORIGINAL) The method of claim 1, wherein the number of load processes is equal to the number of sort processes and which is equal to half of the processing capabilities.

Sub  
B1

12. (ORIGINAL) The method of claim 1, wherein a number of indexes is less than half of the processing capabilities, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that a number of load processes is the smaller of the difference of the processing capabilities available for the load processes and the number of sort processes, and a number of partitions.

13. (ORIGINAL) The method of claim 1, wherein a number of partitions is less than half of the processing capabilities, wherein a number of load processes is equal to the number of partitions, and further comprising determining that a number of sort processes is the smaller of the difference of the processing capabilities available for the sort processes and the number of load processes, and a number of indexes.

14. (ORIGINAL) The method of claim 1, wherein a number of indexes is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of a total amount of available memory and the amount of memory required for the main process, and the amount of memory for each sort process multiplied by the number of indexes, divided by the memory required for each load process, and a number of partitions.

15. (ORIGINAL) The method of claim 1, wherein the number of partitions is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process, wherein a number of load processes is equal to the number of partitions, and further comprising determining that the number of sort processes is the smaller of the difference of the total amount of available memory, the

amount of memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

16. (ORIGINAL) The method of claim 1, wherein a number of load processes is equal to a number of sort processes which is equal to the difference of the total amount of available memory available and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process.

Sub  
B1 }  
17. (ORIGINAL) The method of claim 1, wherein the number of indexes is less than the difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process multiplied by the processing capabilities, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of the processing capabilities and the number of indexes, and a number of partitions.

18. (ORIGINAL) The method of claim 1, wherein the number of partitions is less than the difference of the sum of the amount of memory required for each sort process multiplied by the processing capabilities, the total amount of memory required for a main process, and the amount of memory required for each load process, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of load processes is equal to the number of partitions, and further comprising determining that the number of sort processes is the smaller of the difference of the total amount of available memory, the amount of memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

19. (ORIGINAL) The method of claim 1, wherein a number of sort processes is equal to difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process, divided by the difference of the amount of memory required for each sort process and each load process, and wherein the number of load processes is equal to the difference of the processing capabilities and the number of sort processes.

20. (ORIGINAL) An apparatus for executing parallel load operations, comprising:  
a computer having a data store coupled thereto, wherein the data store stores data; and  
one or more computer programs, performed by the computer, for identifying memory constraints,  
identifying processing capabilities, and determining a number of load and sort processes to be started in  
parallel based on the identified memory constraints and processing capabilities.

#2  
21. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, further comprising  
determining a number of build processes based on the number of sort processes.

Sub  
B1  
22. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of  
sort processes does not exceed a number of indexes to be built.

23. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of  
load processes does not exceed a number of partitions to be loaded.

24. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the total number  
of load and sort processes does not exceed processing capabilities.

25. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the memory  
utilized by the load and sort processes does not exceed memory constraints.

26. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of  
load processes and the number of sort processes each require different processing power.

27. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of  
load processes and the number of sort processes each require similar processing power.

28. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of  
load processes is not equal to the number of sort processes.

29. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of load processes is equal to the number of sort processes.

30. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of load processes is equal to the number of sort processes which is equal to half of the processing capabilities.

31. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein a number of indexes is less than half of the processing capabilities, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that a number of load processes is the smaller of the difference of the processing capabilities available for the load processes and the number of sort processes, and a number of partitions.

32. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein a number of partitions is less than half of the processing capabilities, wherein a number of load processes is equal to the number of partitions, and further comprising determining that a number of sort processes is the smaller of the difference of the processing capabilities available for the sort processes and the number of load processes, and a number of indexes.

33. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein a number of indexes is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of a total amount of available memory and the amount of memory required for the main process, and the amount of memory for each sort process multiplied by the number of indexes, divided by the memory required for each load process, and on a number of partitions.

34. (CURRENTLY AMENDED) The apparatus of claim ~~22~~ 20, wherein the number of partitions is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process is equal to the number of partitions, and further comprising determining that the number of

sort processes is the smaller of the difference of the total amount of available memory, the amount of memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

35. (CURRENTLY AMENDED) The apparatus of claim 22 20, wherein a number of load processes is equal to a number of sort processes which is equal to the difference of the total amount of available memory available and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process.

36. (CURRENTLY AMENDED) The apparatus of claim 22 20, wherein the number of indexes is less than the difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process multiplied by the processing capabilities, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of the processing capabilities and the number of indexes and, a number of partitions.

37. (CURRENTLY AMENDED) The apparatus of claim 22 20, wherein the number of partitions is less than the difference of the sum of the amount of memory required for each sort process multiplied by the processing capabilities, the total amount of memory required for a main process, and the amount of memory required for each load process, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of load processes is equal to the number of partitions, and further comprising determining that the number of sort processes is the smaller of the difference of the total amount of available memory, the amount of memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

38. (CURRENTLY AMENDED) The apparatus of claim 22 20, wherein a number of sort processes is equal to difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process, divided by the

Cont  
A2

difference of the amount of memory required for each sort process and each load process, and wherein the number of load processes is equal to the difference to the processing capabilities and the number of sort processes.

Sub  
B1

39. (ORIGINAL) An article of manufacture comprising a program storage medium readable by a computer and embodying one or more instructions executable by the computer to perform method steps for loading data into a data store connected to a computer, the method comprising the steps of:

- identifying memory constraints;
- identifying processing capabilities; and
- determining a number of load and sort processes to be started in parallel based on the identified memory constraints and processing capabilities.

A3

40. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, further comprising determining a number of build processes based on the number of sort processes.

41. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of sort processes does not exceed a number of indexes to be built.

42. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes does not exceed a number of partitions to be loaded.

43. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the total number of load and sort processes does not exceed processing capabilities.

44. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the memory utilized by the load and sort processes does not exceed memory constraints.

45. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes and the number of sort processes each require different processing power.

46. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes and the number of sort processes each require similar processing power.

47. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes is not equal to the number of sort processes.

48. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes is equal to the number of sort processes.

49. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of load processes is equal to the number of sort processes which is equal to half of the processing capabilities.

50. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein a number of indexes is less than half of the processing capabilities, wherein a number of sort processes is equal to the number of sort processes is equal to the number of indexes, and further comprising, determining that a number of load processes is the smaller of the difference of the processing capabilities available for the load processes and the number of sort processes, and a number of partitions.

51. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein a number of partitions is less than half of the processing capabilities, wherein a number of load processes is equal to the number of partitions, and further comprising determining that a number of sort processes is the smaller of the difference of the processing capabilities available for the sort processes and the number of load processes, and on a number of indexes.

52. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein a number of indexes is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of a total amount of available memory and the amount of memory required for the main process, and the



amount of memory for each sort process multiplied by the number of indexes, divided by the memory required for each load process and, a number of partitions.

53. (CURRENTLY AMENDED) The article of manufacture claim 43 39, wherein the number of partitions is less than the difference of the total amount of available memory and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process, wherein a number of load processes is equal to the number of partitions, and further comprising determining that the number of sort processes is the smaller of the difference of the total amount of available memory, the amount of memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

54. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein a number of load processes is equal to a number of sort processes which is equal to the difference of the total amount of available memory available and the amount of memory required for a main process, divided by the amount of memory required for each load and sort process.

55. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of indexes is less than the difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process multiplied by the processing capabilities, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of sort processes is equal to the number of indexes, and further comprising determining that the number of load processes is the smaller of the difference of the processing capabilities and the number of indexes and, a number of partitions.

56. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein the number of partitions is less than the difference of the sum of the amount of memory required for each sort process multiplied by the processing capabilities, the total amount of memory required for a main process, and the amount of memory required for each load process, divided by the difference of the amount of memory required for each sort process and each load process, wherein a number of load processes is equal to the number of partitions, and further comprising determining that the number of sort processes is the smaller of the difference of the total amount of available memory, the amount of

memory required for the main process, and the amount of memory for each load process multiplied by the number of partitions, divided by the memory required for each sort process, and a number of indexes.

57. (CURRENTLY AMENDED) The article of manufacture of claim 43 39, wherein a number of sort processes is equal to difference of the total amount of available memory, the amount of memory required for a main process, and the amount of memory required for each load process, divided by the difference of the amount of memory required for each sort process and each load process, and wherein the number of load processes is equal to the difference of the processing capabilities and the number of sort processes.